Configuração Loraserver 1.0.1

A instalação do loraserver é direta e simples. Para este projeto a instalação e execução é feita via docker seguindo as orientações do tutorial provido pelo loraserver.io

https://www.loraserver.io/install/docker/

```
$ git clone https://github.com/brocaar/loraserver-docker.git
$ cd loraserver-docker
$ docker-compose up
```

`Tutorial instalação DOCKER`

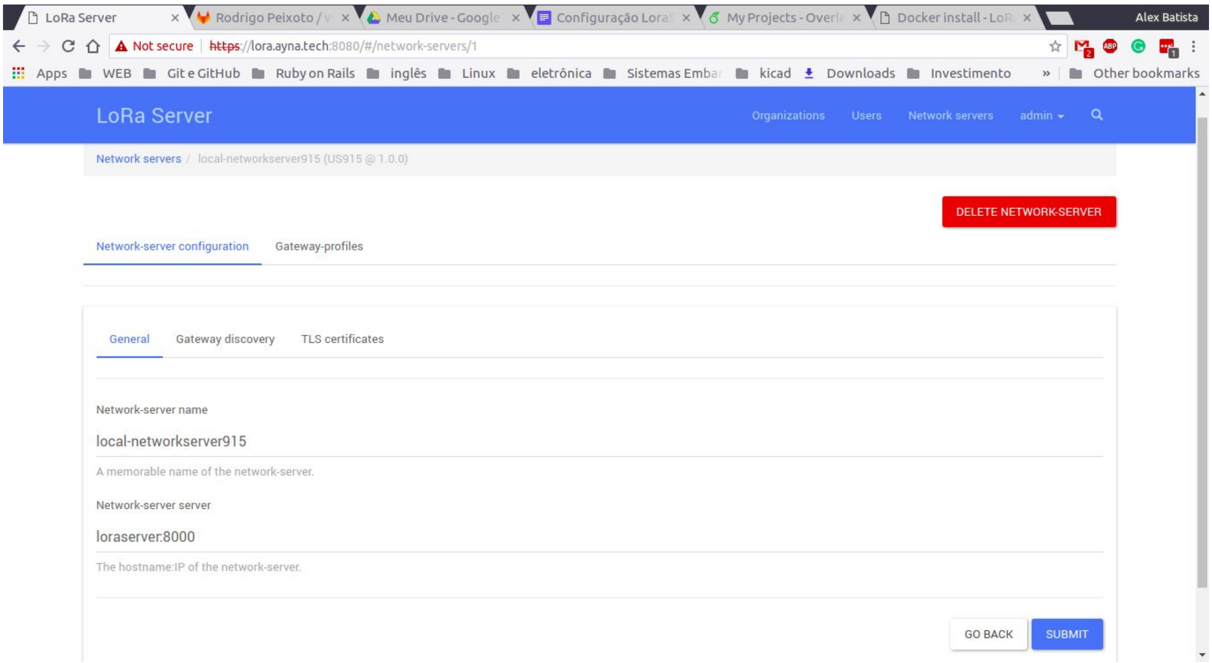
```
Versão da Docker utilizada: Docker version 17.05.0-ce, build 89658be
```

`Tutorial instalação DOCKER COMPOSE`

```
Versão da Docker-Compose utilizada: docker-compose version 1.21.2,
build a133471
```

A seguir estão as configurações utilizadas na instância do loraserver em produção:

Crie uma nova conexão com o networkserver local através de seu endereço loraserver:8000. Este networkserver sobe junto com a docker.



Network servers / local-networkserver915 (US915 @ 1.0.0)

Network-server name

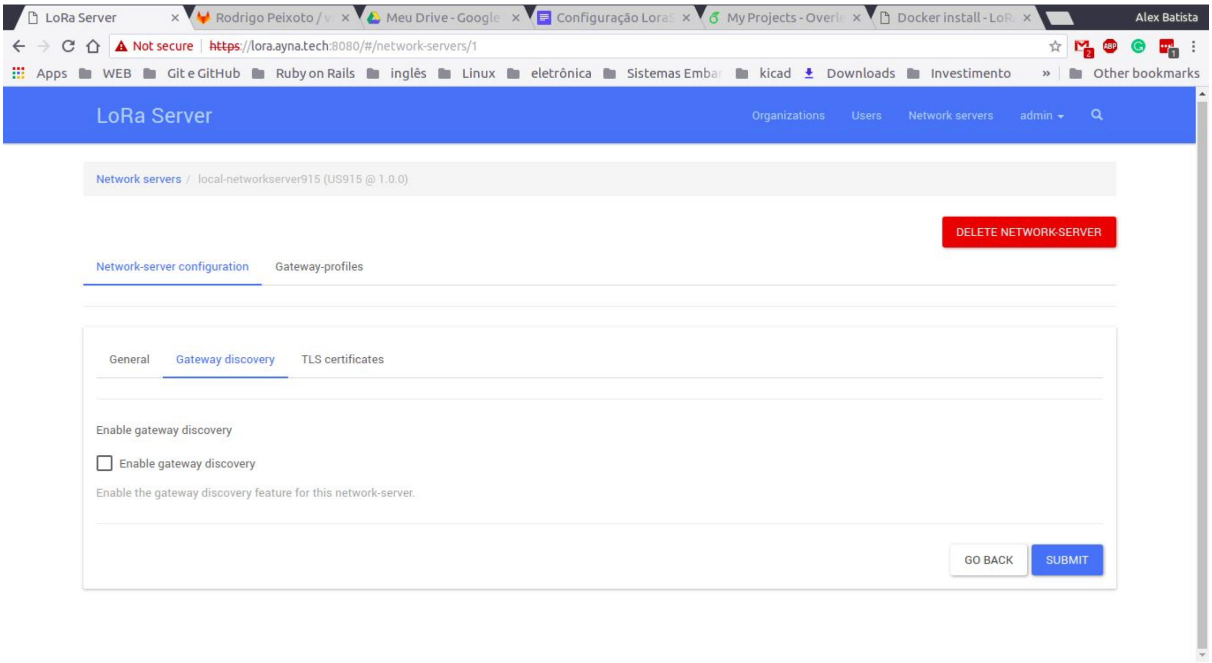
local-networkserver915

A memorable name of the network-server.

Network-server server

loraserver:8000

The hostname:IP of the network-server.



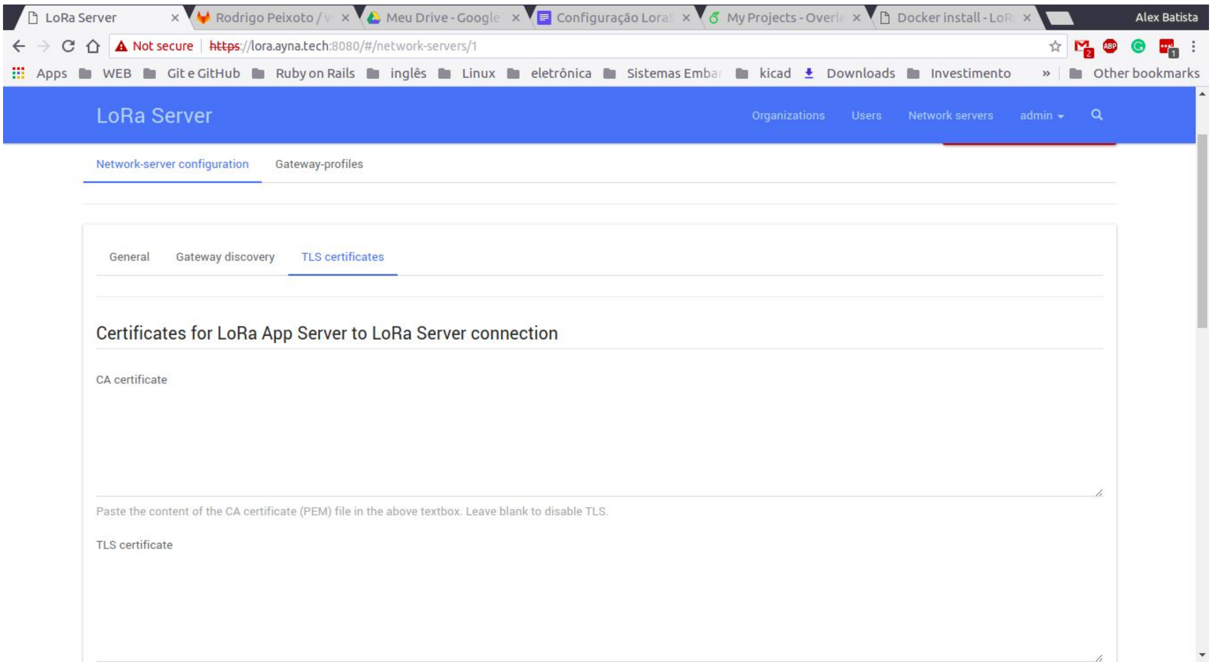
Network servers / local-networkserver915 (US915 @ 1.0.0)

Enable gateway discovery

□ Enable gateway discovery

Enable the gateway discovery feature for this network-server.

Não foram usados certificados



https://lora.ayna.tech:8080/#/network-servers/1

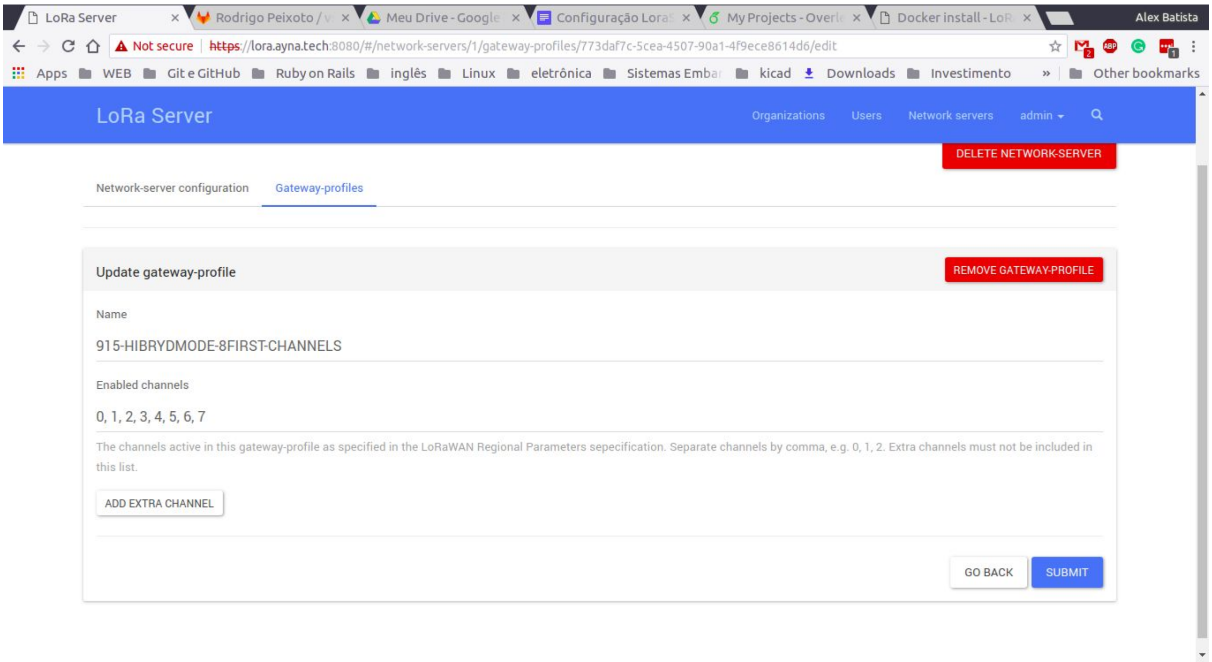
Certificates for LoRa App Server to LoRa Server connection

CA certificate

Paste the content of the CA certificate (PEM) file in the above textbox. Leave blank to disable TLS.

TLS certificate

Crie um novo gateway-profile onde será configurado o uso dos 8 primeiros canais conforme especificado para US915 modo híbrido. Os nós já estão configurados para utilizarem os 8 primeiros canais. De igual modo, os gateways foram configurados para utilizarem os 8 primeiros canais. Durante os primeiros testes tudo funcionou bem sem esta configuração, ela precisa ser melhor entendida, mas os canais de 0 a 7 estão corretos.



https://lora.ayna.tech:8080/#/network-servers/1/gateway-profiles/773daf7c-5cea-4507-90a1-4f9ece8614d6/edit

Update gateway-profile

Name

915-HIBRYDMODE-8FIRST-CHANNELS

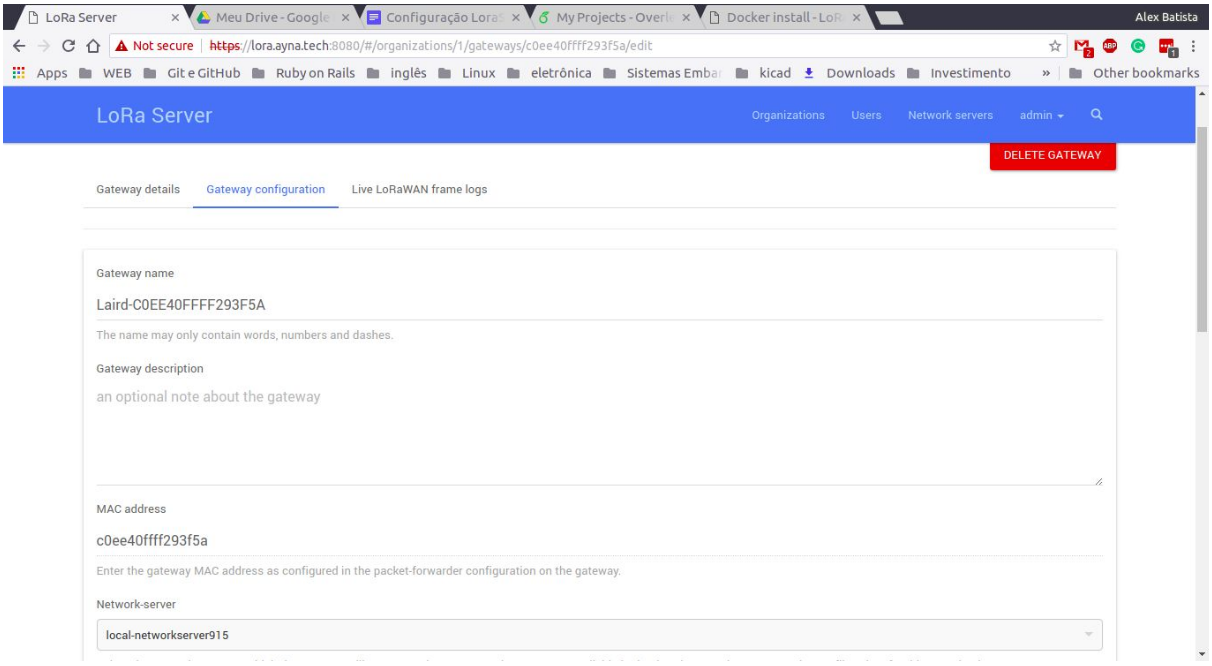
Enabled channels

0, 1, 2, 3, 4, 5, 6, 7

The channels active in this gateway-profile as specified in the LoRaWAN Regional Parameters sepecification. Separate channels by comma, e.g. 0, 1, 2. Extra channels must not be included in this list.

Configurações dos gateways registrados

Os gateways são configurados utilizando o gateway profile criado anteriormente onde são setados os 8 primeiros canais.



Gateway name

Laird-C0EE40FFFF293F5A

The name may only contain words, numbers and dashes.

Gateway description

an optional note about the gateway

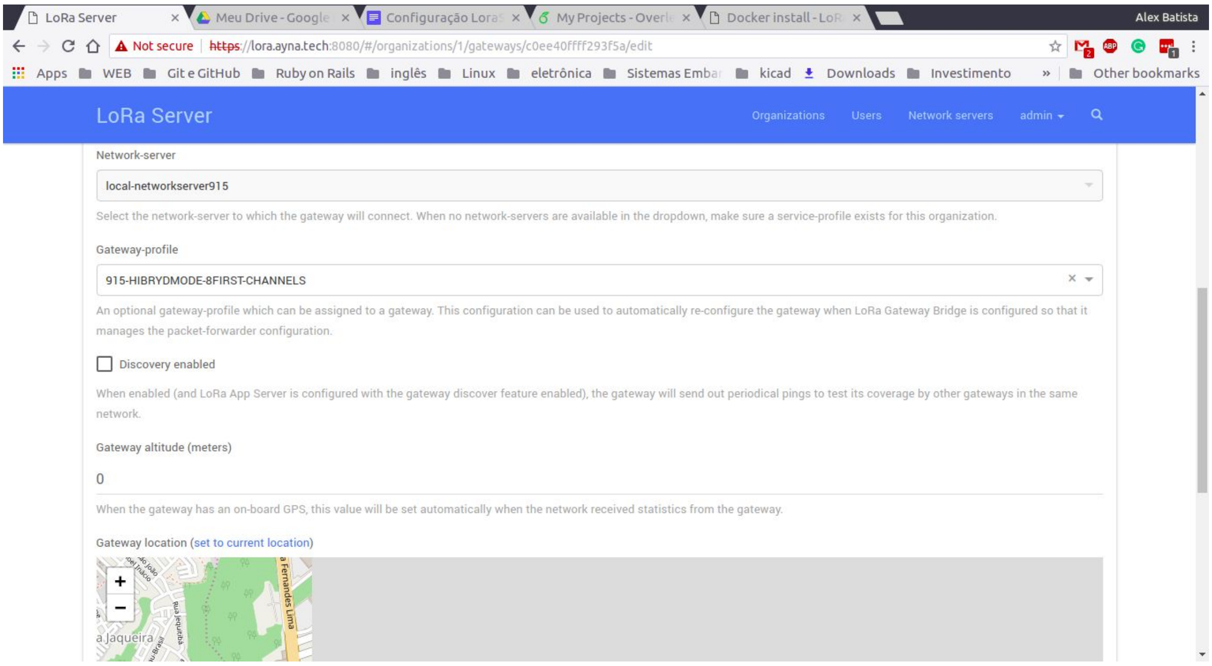
MAC address

c0ee40ffff293f5a

Enter the gateway MAC address as configured in the packet-forwarder configuration on the gateway.

Network-server

local-networkserver915



Network-server

local-networkserver915

Select the network-server to which the gateway will connect. When no network-servers are available in the dropdown, make sure a service-profile exists for this organization.

Gateway-profile

915-HIBRYDMODE-8FIRST-CHANNELS

An optional gateway-profile which can be assigned to a gateway. This configuration can be used to automatically re-configure the gateway when LoRa Gateway Bridge is configured so that it manages the packet-forwarder configuration.

☐ Discovery enabled

When enabled (and LoRa App Server is configured with the gateway discover feature enabled), the gateway will send out periodical pings to test its coverage by other gateways in the same network.

Gateway altitude (meters)

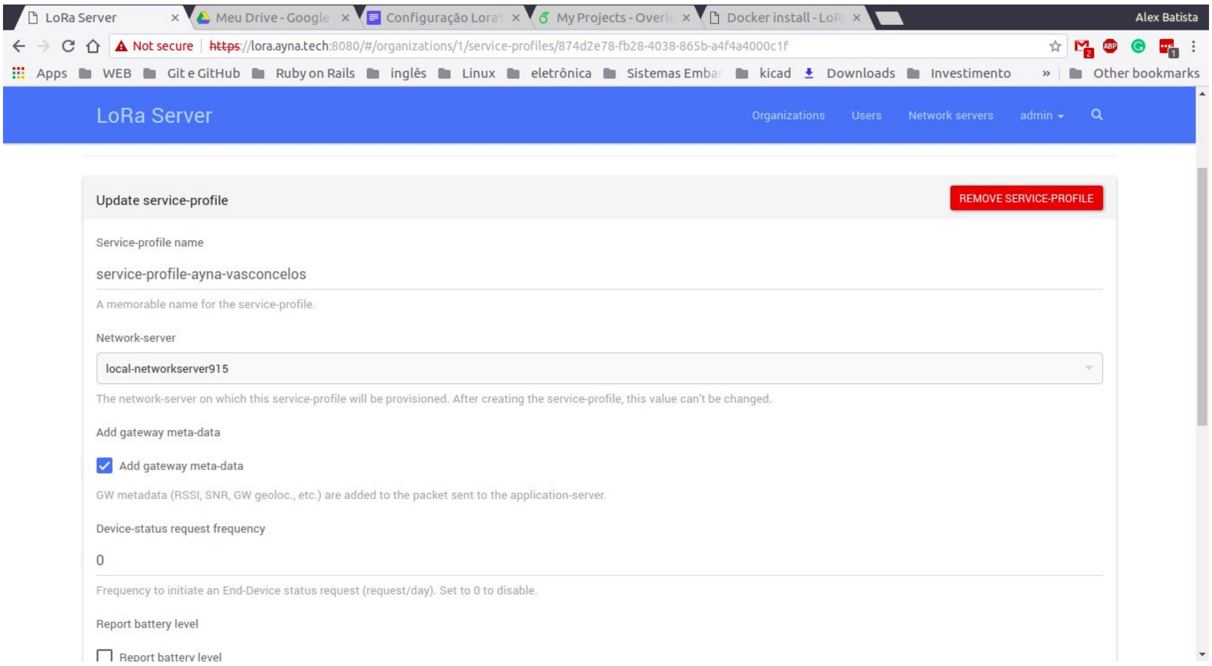
0

When the gateway has an on-board GPS, this value will be set automatically when the network received statistics from the gateway.

Gateway location (set to current location)

SERVICE PROFILES

O service profile criado utiliza o networkserver configurado anteriormente e define o data-rate mínimo de 0 e máximo de 3.



Update service-profile

Service-profile name

service-profile-ayna-vasconcelos

A memorable name for the service-profile.

Network-server

local-networkserver915

The network-server on which this service-profile will be provisioned. After creating the service-profile, this value can't be changed.

Add gateway meta-data

☑ Add gateway meta-data

GW metadata (RSSI, SNR, GW geoloc., etc.) are added to the packet sent to the application-server.

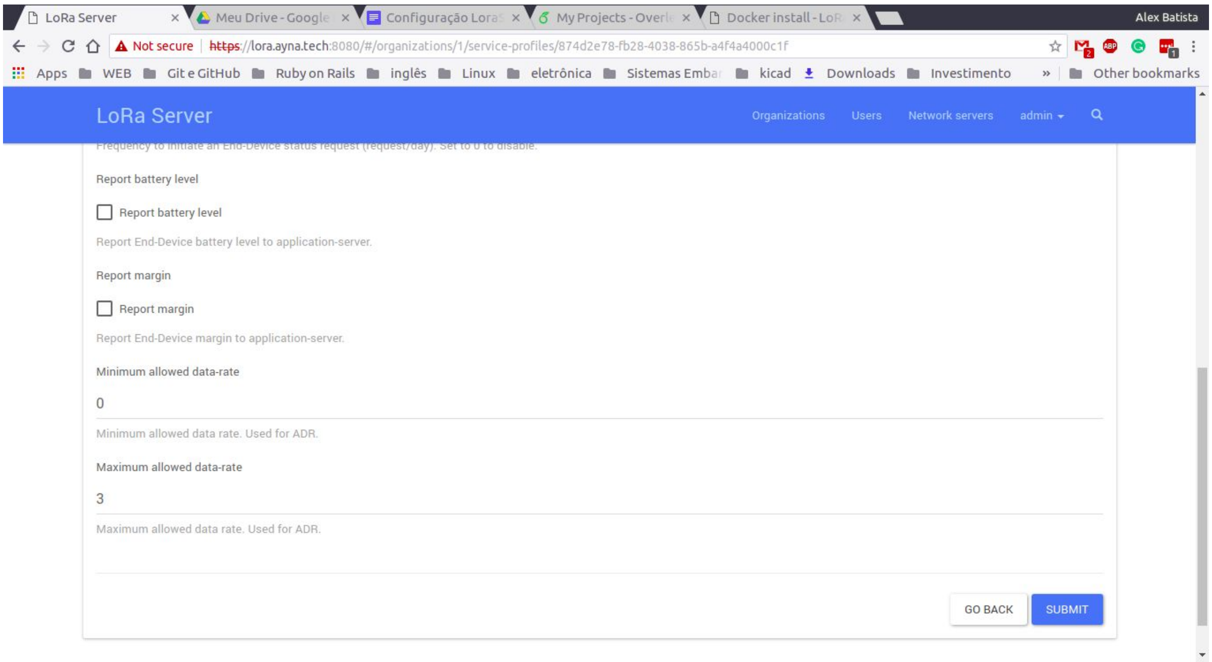
Device-status request frequency

0

Frequency to initiate an End-Device status request (request/day). Set to 0 to disable.

Report battery level

□ Report battery level



Frequency to initiate an End-Device status request (request/day). Set to 0 to disable.

Report battery level

□ Report battery level

Report End-Device battery level to application-server.

Report margin

□ Report margin

Report End-Device margin to application-server.

Minimum allowed data-rate

0

Minimum allowed data rate. Used for ADR.

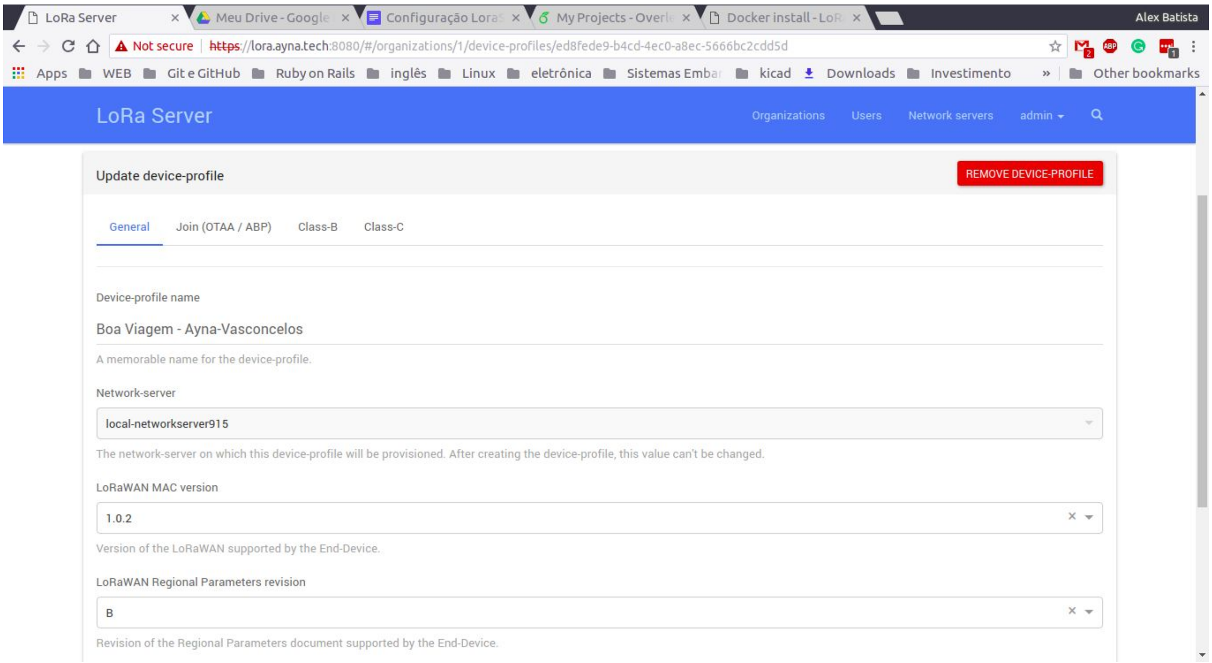
Maximum allowed data-rate

3

Maximum allowed data rate. Used for ADR.

DEVICE PROFILES

Device profiles configurados utilizando a versão loramac 1.0.2 revision B, esta é a versão do código utilizado no firmware.



Update device-profile

Device-profile name

Boa Viagem - Ayna-Vasconcelos

A memorable name for the device-profile.

Network-server

local-networkserver915

The network-server on which this device-profile will be provisioned. After creating the device-profile, this value can't be changed.

LoRaWAN MAC version

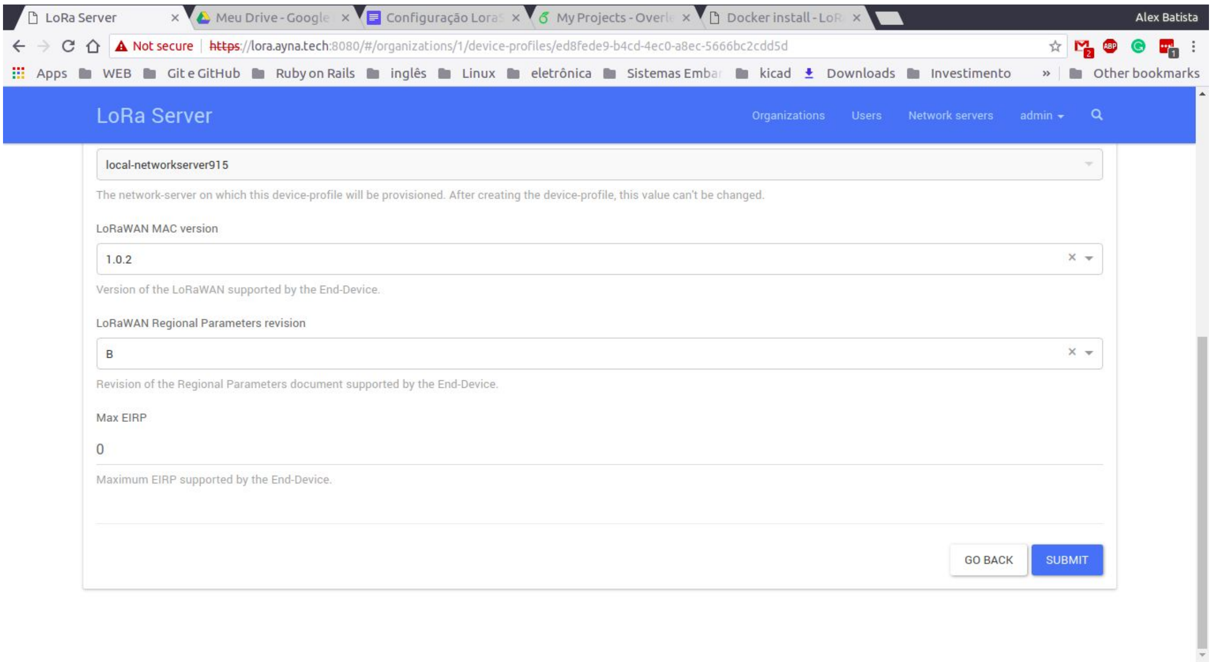
1.0.2

Version of the LoRaWAN supported by the End-Device.

LoRaWAN Regional Parameters revision

B

Revision of the Regional Parameters document supported by the End-Device.



local-networkserver915

The network-server on which this device-profile will be provisioned. After creating the device-profile, this value can't be changed.

LoRaWAN MAC version

1.0.2

Version of the LoRaWAN supported by the End-Device.

LoRaWAN Regional Parameters revision

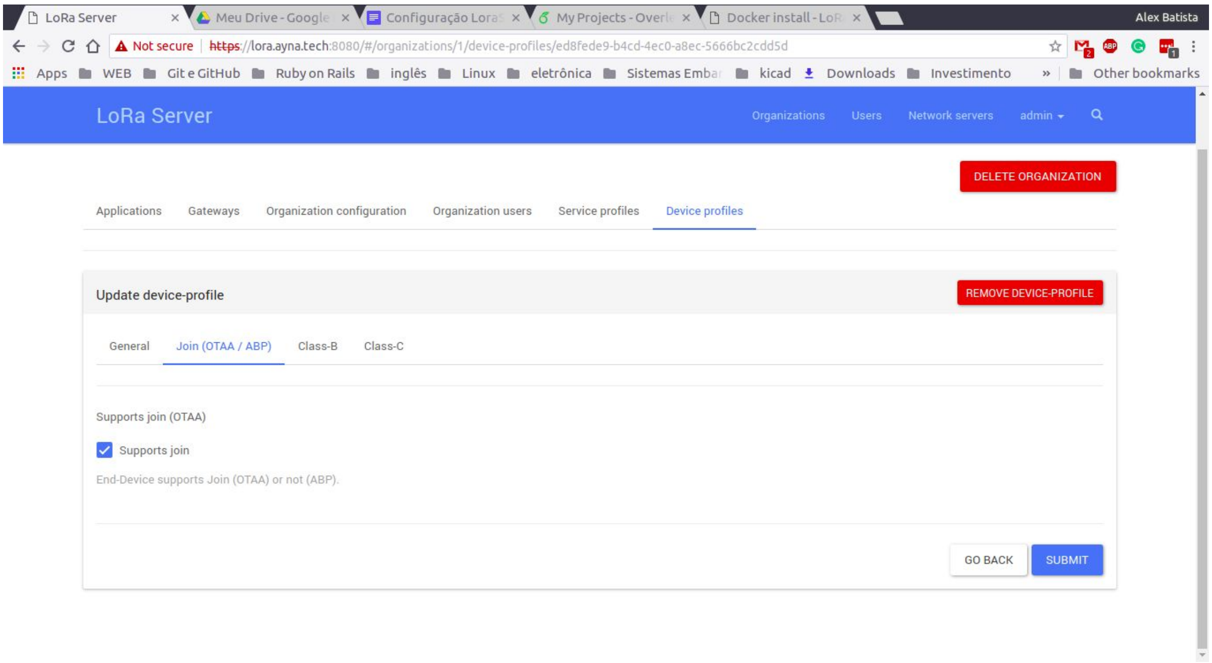
B

Revision of the Regional Parameters document supported by the End-Device.

Max EIRP

0

Maximum EIRP supported by the End-Device.



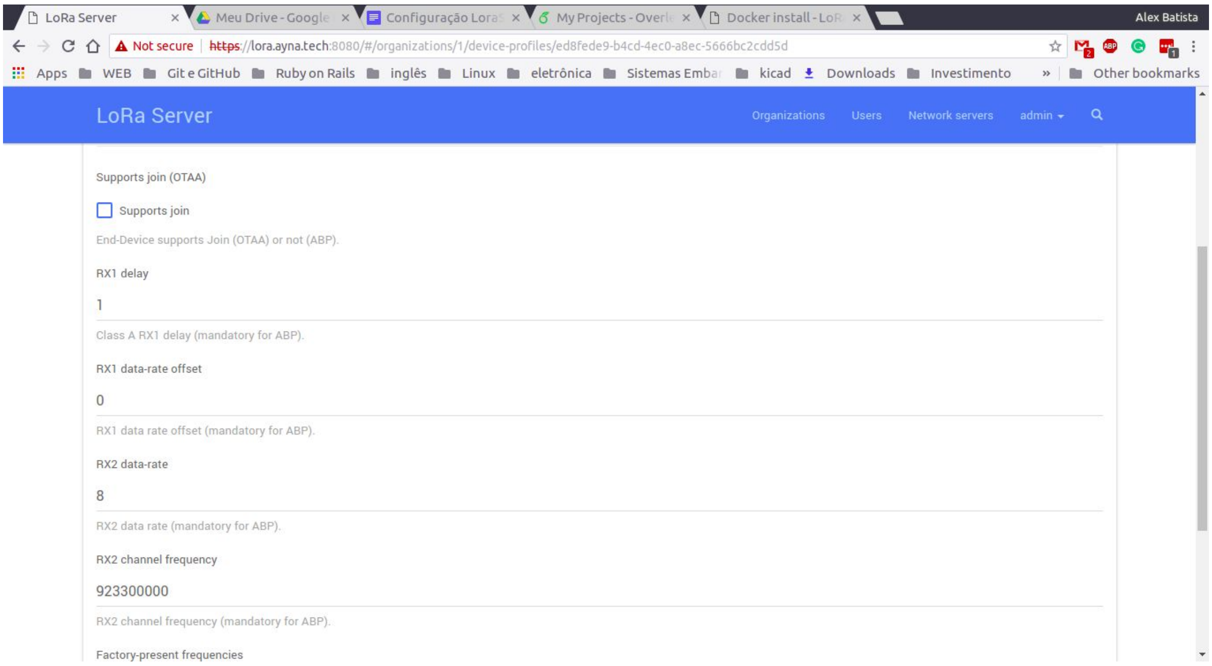
Update device-profile

Supports join (OTAA)

☑ Supports join

End-Device supports Join (OTAA) or not (ABP).

A opção *supports join* deve estar marcada, o print abaixo é apenas uma referência do que tinha posto antes, por via das dúvidas deixe preenchido dessa forma e depois marque a opção *supports join*:



Supports join (OTAA)

□ Supports join

End-Device supports Join (OTAA) or not (ABP).

RX1 delay

1

Class A RX1 delay (mandatory for ABP).

RX1 data-rate offset

0

RX1 data rate offset (mandatory for ABP).

RX2 data-rate

8

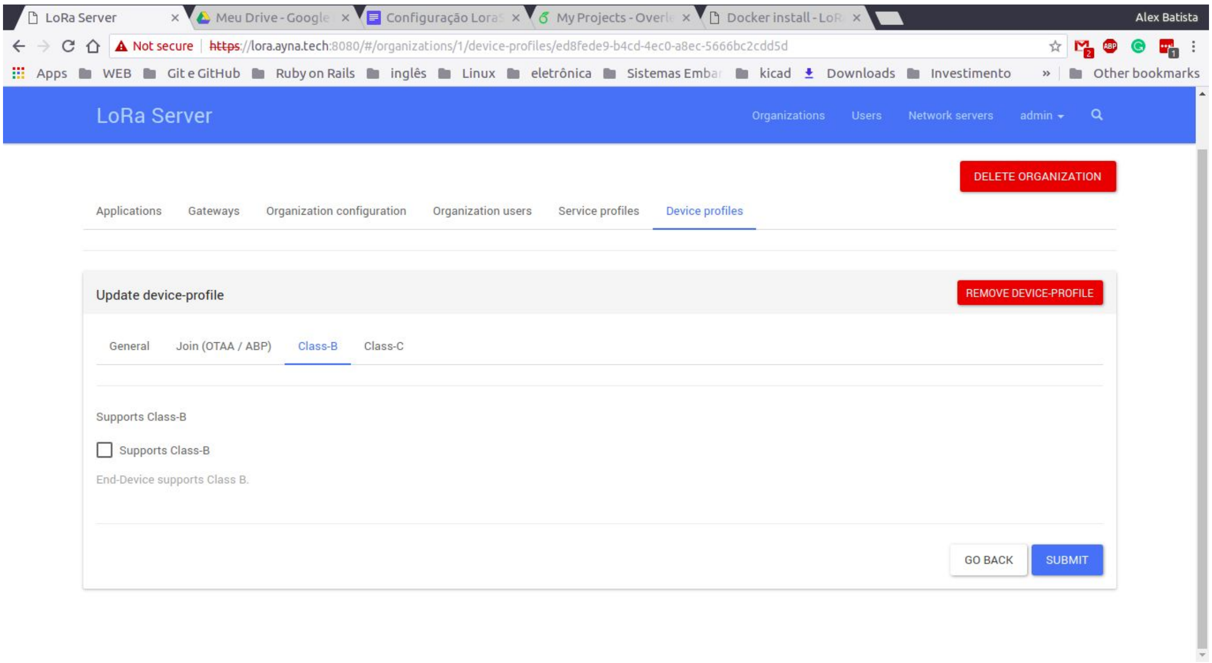
RX2 data rate (mandatory for ABP).

RX2 channel frequency

923300000

RX2 channel frequency (mandatory for ABP).

Factory-present frequencies

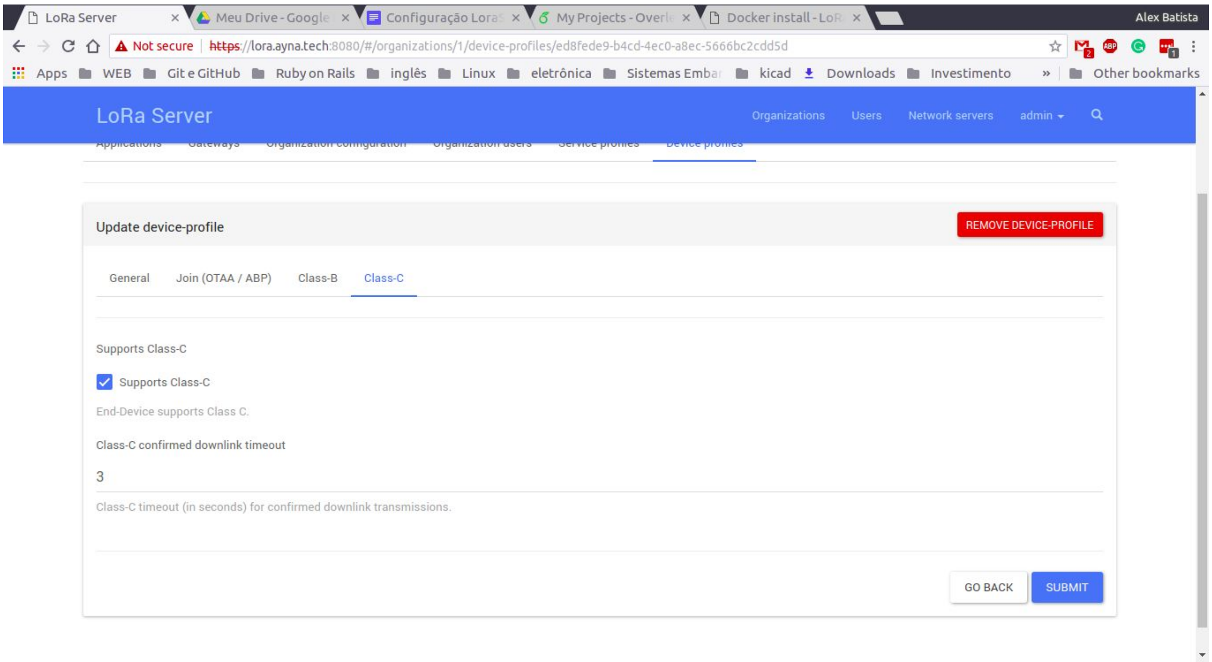


Update device-profile

Supports Class-B

□ Supports Class-B

End-Device supports Class B.



Update device-profile

Supports Class-C

☑ Supports Class-C

End-Device supports Class C.

Class-C confirmed downlink timeout

3

Class-C timeout (in seconds) for confirmed downlink transmissions.

Configurações utilizadas nos arquivos da docker:

Na instância do EC2, em loraserver-docker/configuration/lora-app-server as seguintes configurações foram utilizadas no arquivo lora-app-server.toml:

```
[general]
# Log level
#
# debug=5, info=4, warning=3, error=2, fatal=1, panic=0
```

```
log_level=4

# The number of times passwords must be hashed. A higher number is safer as
# an attack takes more time to perform.
password_hash_iterations=100000


# PostgreSQL settings.
#
# Please note that PostgreSQL 9.5+ is required.
[postgresql]
# PostgreSQL dsn (e.g.:
postgres://user:password@hostname/database?sslmode=disable).
#
# Besides using an URL (e.g.
'postgres://user:password@hostname/database?sslmode=disable')
# it is also possible to use the following format:
# 'user=loraserver dbname=loraserver sslmode=disable'.
#
# The following connection parameters are supported:
#
# * dbname - The name of the database to connect to
# * user - The user to sign in as
# * password - The user's password
# * host - The host to connect to. Values that start with / are for unix domain sockets.
(default is localhost)
# * port - The port to bind to. (default is 5432)
# * sslmode - Whether or not to use SSL (default is require, this is not the default for libpq)
# * fallback_application_name - An application_name to fall back to if one isn't provided.
# * connect_timeout - Maximum wait for connection, in seconds. Zero or not specified
means wait indefinitely.
# * sslcert - Cert file location. The file must contain PEM encoded data.
# * sslkey - Key file location. The file must contain PEM encoded data.
# * sslrootcert - The location of the root certificate file. The file must contain PEM encoded
data.
#
# Valid values for sslmode are:
#
# * disable - No SSL
# * require - Always SSL (skip verification)
# * verify-ca - Always SSL (verify that the certificate presented by the server was signed
by a trusted CA)
# * verify-full - Always SSL (verify that the certification presented by the server was signed
by a trusted CA and the server host name matches the one in the certificate)
dsn="postgres://loraserver_as:loraserver_as@postgresql/loraserver_as?sslmode=disable"

# Automatically apply database migrations.
#
# It is possible to apply the database-migrations by hand
# (see https://github.com/brocaar/lora-app-server/tree/master/migrations)
# or let LoRa App Server migrate to the latest state automatically, by using
# this setting. Make sure that you always make a backup when upgrading Lora
# App Server and / or applying migrations.
automigrate=true


# Redis settings
#
# Please note that Redis 2.6.0+ is required.
[redis]
# Redis url (e.g. redis://user:password@hostname/0)
#
# For more information about the Redis URL format, see:
# https://www.iana.org/assignments/uri-schemes/prov/redis
url="redis://redis:6379"


# Application-server settings.
[application_server]
# Application-server identifier.
#
# Random UUID defining the id of the application-server installation (used by
# LoRa Server as routing-profile id).
# For now it is recommended to not change this id.
id="6d5db27e-4ce2-4b2b-b5d7-91f069397978"


  # MQTT integration configuration used for publishing (data) events
  # and scheduling downlink application payloads.
  # Next to this integration which is always available, the user is able to
  # configure additional per-application integrations.
  [application_server.integration.mqtt]
  # MQTT topic templates for the different MQTT topics.
  #
  # The meaning of these topics are documented at:
  # https://docs.loraserver.io/lora-app-server/integrate/data/
  #
  # The following substitutions can be used:
  # * "{{ .ApplicationID }}" for the application id.
  # * "{{ .DevEUI }}" for the DevEUI of the device.
  #
  # Note: the downlink_topic_template must contain both the application id and
  # DevEUI substitution!
  uplink_topic_template="application/{{ .ApplicationID }}/node/{{ .DevEUI }}/rx"
  downlink_topic_template="application/{{ .ApplicationID }}/node/{{ .DevEUI }}/tx"
  join_topic_template="application/{{ .ApplicationID }}/node/{{ .DevEUI }}/join"
  ack_topic_template="application/{{ .ApplicationID }}/node/{{ .DevEUI }}/ack"
  error_topic_template="application/{{ .ApplicationID }}/node/{{ .DevEUI }}/error"

  # MQTT server (e.g. scheme://host:port where scheme is tcp, ssl or ws)
```

```
 server="tcp://mosquitto:1883"

 # Connect with the given username (optional)
 username=""

 # Connect with the given password (optional)
 password=""

 # Quality of service level
 #
 # 0: at most once
 # 1: at least once
 # 2: exactly once
 #
 # Note: an increase of this value will decrease the performance.
 # For more information:
https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels
 qos=0

 # Clean session
 #
 # Set the "clean session" flag in the connect message when this client
 # connects to an MQTT broker. By setting this flag you are indicating
 # that no messages saved by the broker for this client should be delivered.
 clean_session=true

 # Client ID
 #
 # Set the client id to be used by this client when connecting to the MQTT
 # broker. A client id must be no longer than 23 characters. When left blank,
 # a random id will be generated. This requires clean_session=true.
 client_id=""

 # CA certificate file (optional)
 #
 # Use this when setting up a secure connection (when server uses ssl://...)
 # but the certificate used by the server is not trusted by any CA certificate
 # on the server (e.g. when self generated).
 ca_cert=""

 # TLS certificate file (optional)
 tls_cert=""

 # TLS key file (optional)
 tls_key=""


 # Settings for the "internal api"
 #
 # This is the API used by LoRa Server to communicate with LoRa App Server
```

```
  # and should not be exposed to the end-user.
  [application_server.api]
  # ip:port to bind the api server
  bind="0.0.0.0:8001"

  # ca certificate used by the api server (optional)
  ca_cert=""

  # tls certificate used by the api server (optional)
  tls_cert=""

  # tls key used by the api server (optional)
  tls_key=""

  # Public ip:port of the application-server API.
  #
  # This is used by LoRa Server to connect to LoRa App Server. When running
  # LoRa App Server on a different host than LoRa Server, make sure to set
  # this to the host:ip on which LoRa Server can reach LoRa App Server.
  # The port must be equal to the port configured by the 'bind' flag
  # above.
  public_host="appserver:8001"


  # Settings for the "external api"
  #
  # This is the API and web-interface exposed to the end-user.
  [application_server.external_api]
  # ip:port to bind the (user facing) http server to (web-interface and REST / gRPC api)
  bind="0.0.0.0:8080"

  # http server TLS certificate
  tls_cert="/etc/lora-app-server/certs/http.pem"

  # http server TLS key
  tls_key="/etc/lora-app-server/certs/http-key.pem"

  # JWT secret used for api authentication / authorization
  # You could generate this by executing 'openssl rand -base64 32' for example
  jwt_secret="verysecret"

  # when set, existing users can't be re-assigned (to avoid exposure of all users to an
organization admin)"
  disable_assign_existing_users=false



# Join-server configuration.
#
# LoRa App Server implements a (subset) of the join-api specified by the
# LoRaWAN Backend Interfaces specification. This API is used by LoRa Server
# to handle join-requests.
[join_server]
# ip:port to bind the join-server api interface to
bind="0.0.0.0:8003"

# ca certificate used by the join-server api server
ca_cert=""

# tls certificate used by the join-server api server (optional)
tls_cert=""

# tls key used by the join-server api server (optional)
tls_key=""


# Network-server configuration.
#
# This configuration is only used to migrate from older LoRa App Server.
[network_server]
server="127.0.0.1:8000"
```

Na instância do EC2, em loraserver-docker/configuration/lora-gateway-bridge as seguintes configurações foram utilizadas no arquivo lora-gateway-bridge.toml:

```
[general]
# debug=5, info=4, warning=3, error=2, fatal=1, panic=0
log_level = 4


# Configuration which relates to the packet-forwarder.
[packet_forwarder]
# ip:port to bind the UDP listener to
#
# Example: 0.0.0.0:1700 to listen on port 1700 for all network interfaces.
# This is the listeren to which the packet-forwarder forwards its data
# so make sure the 'serv_port_up' and 'serv_port_down' from your
# packet-forwarder matches this port.
udp_bind = "0.0.0.0:1700"

# Skip the CRC status-check of received packets
#
# This is only has effect when the packet-forwarder is configured to forward
# LoRa frames with CRC errors.
skip_crc_check = false


# Configuration for the MQTT backend.
[backend.mqtt]
# MQTT topic templates for the different MQTT topics.
#
# The meaning of these topics are documented at:
# https://docs.loraserver.io/lora-gateway-bridge/use/data/
#
# The default values match the default expected configuration of the
# LoRa Server MQTT backend. Therefore only change these values when
# absolutely needed.
# Use "{{ .MAC }}" as an substitution for the LoRa gateway MAC.
uplink_topic_template="gateway/{{ .MAC }}/rx"
downlink_topic_template="gateway/{{ .MAC }}/tx"
stats_topic_template="gateway/{{ .MAC }}/stats"
ack_topic_template="gateway/{{ .MAC }}/ack"

# MQTT server (e.g. scheme://host:port where scheme is tcp, ssl or ws)
server="tcp://mosquitto:1883"

# Connect with the given username (optional)
username=""

# Connect with the given password (optional)
password=""

# Quality of service level
#
# 0: at most once
# 1: at least once
# 2: exactly once
#
# Note: an increase of this value will decrease the performance.
# For more information:
https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels
qos=0

# Clean session
#
# Set the "clean session" flag in the connect message when this client
# connects to an MQTT broker. By setting this flag you are indicating
# that no messages saved by the broker for this client should be delivered.
clean_session=true

# Client ID
#
# Set the client id to be used by this client when connecting to the MQTT
# broker. A client id must be no longer than 23 characters. When left blank,
# a random id will be generated. This requires clean_session=true.
client_id=""

# CA certificate file (optional)
#
# Use this when setting up a secure connection (when server uses ssl://...)
# but the certificate used by the server is not trusted by any CA certificate
# on the server (e.g. when self generated).
ca_cert=""

# mqtt TLS certificate file (optional)
tls_cert=""

# mqtt TLS key file (optional)
tls_key=""
```

Na instância do EC2, em loraserver-docker/configuration/loraserver as seguintes configurações foram utilizadas no arquivo loraserver.toml:

```
[general]
# Log level
#
# debug=5, info=4, warning=3, error=2, fatal=1, panic=0
log_level=4


# PostgreSQL settings.
#
# Please note that PostgreSQL 9.5+ is required.
[postgresql]
# PostgreSQL dsn (e.g.:
postgres://user:password@hostname/database?sslmode=disable).
#
# Besides using an URL (e.g.
'postgres://user:password@hostname/database?sslmode=disable')
# it is also possible to use the following format:
# 'user=loraserver dbname=loraserver sslmode=disable'.
#
# The following connection parameters are supported:
#
# * dbname - The name of the database to connect to
# * user - The user to sign in as
# * password - The user's password
# * host - The host to connect to. Values that start with / are for unix domain sockets.
```

```
(default is localhost)
# * port - The port to bind to. (default is 5432)
# * sslmode - Whether or not to use SSL (default is require, this is not the default for libpq)
# * fallback_application_name - An application_name to fall back to if one isn't provided.
# * connect_timeout - Maximum wait for connection, in seconds. Zero or not specified
means wait indefinitely.
# * sslcert - Cert file location. The file must contain PEM encoded data.
# * sslkey - Key file location. The file must contain PEM encoded data.
# * sslrootcert - The location of the root certificate file. The file must contain PEM encoded
data.
#
# Valid values for sslmode are:
#
# * disable - No SSL
# * require - Always SSL (skip verification)
# * verify-ca - Always SSL (verify that the certificate presented by the server was signed
by a trusted CA)
# * verify-full - Always SSL (verify that the certification presented by the server was signed
by a trusted CA and the server host name matches the one in the certificate)
dsn="postgres://loraserver_ns:loraserver_ns@postgresql/loraserver_ns?sslmode=disable"

# Automatically apply database migrations.
#
# It is possible to apply the database-migrations by hand
# (see https://github.com/brocaar/loraserver/tree/master/migrations)
# or let LoRa App Server migrate to the latest state automatically, by using
# this setting. Make sure that you always make a backup when upgrading Lora
# App Server and / or applying migrations.
automigrate=true


# Redis settings
#
# Please note that Redis 2.6.0+ is required.
[redis]
# Redis url (e.g. redis://user:password@hostname/0)
#
# For more information about the Redis URL format, see:
# https://www.iana.org/assignments/uri-schemes/prov/redis
url="redis://redis:6379"


# Network-server settings.
[network_server]
# Network identifier (NetID, 3 bytes) encoded as HEX (e.g. 010203)
net_id="010203"

# Time to wait for uplink de-duplication.
#
# This is the time that LoRa Server will wait for other gateways to receive
# the same uplink frame. Valid units are 'ms' or 's'.
# Please note that this value has influence on the uplink / downlink
# roundtrip time. Setting this value too high means LoRa Server will be
# unable to respond to the device within its receive-window.
deduplication_delay="200ms"

# Device session expiration.
#
# The TTL value defines the time after which a device-session expires
# after no activity. Valid units are 'ms', 's', 'm', 'h'. Note that these
# values can be combined, e.g. '24h30m15s'.
device_session_ttl="744h0m0s"

# Get downlink data delay.
#
# This is the time that LoRa Server waits between forwarding data to the
# application-server and reading data from the queue. A higher value
# means that the application-server has more time to schedule a downlink
# queue item which can be processed within the same uplink / downlink
# transaction.
# Please note that this value has influence on the uplink / downlink
# roundtrip time. Setting this value too high means LoRa Server will be
# unable to respond to the device within its receive-window.
get_downlink_data_delay="100ms"


  # LoRaWAN regional band configuration.
  #
  # Note that you might want to consult the LoRaWAN Regional Parameters
  # specification for valid values that apply to your region.
  # See: https://www.lora-alliance.org/lorawan-for-developers
  [network_server.band]
  # LoRaWAN band to use.
  #
  # Valid values are:
  # *     AS_923
  # * AU_915_928
  # * CN_470_510
  # * CN_779_787
  # * EU_433
  # * EU_863_870
  # * IN_865_867
  # * KR_920_923
  # * RU_864_870
  # * US_902_928
  name="US_902_928"

  # Enforce 400ms dwell time
  #
  # Some band configurations define the max payload size for both dwell-time
  # limitation enabled as disabled (e.g. AS 923). In this case the
  # dwell time setting must be set to enforce the max payload size
  # given the dwell-time limitation. For band configuration where the dwell-time is
  # always enforced, setting this flag is not required.
  dwell_time_400ms=false

  # Enforce repeater compatibility
  #
  # Most band configurations define the max payload size for both an optional
  # repeater encapsulation layer as for setups where a repeater will never
  # be used. The latter case increases the max payload size for some data-rates.
  # In case a repeater might used, set this flag to true.
  repeater_compatible=false


  # LoRaWAN network related settings.
  [network_server.network_settings]
  # Installation margin (dB) used by the ADR engine.
  #
  # A higher number means that the network-server will keep more margin,
  # resulting in a lower data-rate but decreasing the chance that the
  # device gets disconnected because it is unable to reach one of the
  # surrounded gateways.
  installation_margin=10

  # Class A RX1 delay
  #
  # 0=1sec, 1=1sec, ... 15=15sec. A higher value means LoRa Server has more
  # time to respond to the device as the delay between the uplink and the
  # first receive-window will be increased.
  rx1_delay=1

  # RX1 data-rate offset
  #
  # Please consult the LoRaWAN Regional Parameters specification for valid
  # options of the configured network_server.band.name.
  rx1_dr_offset=0

  # RX2 data-rate
  #
  # When set to -1, the default RX2 data-rate will be used for the configured
  # LoRaWAN band.
  #
  # Please consult the LoRaWAN Regional Parameters specification for valid
  # options of the configured network_server.band.name.
  rx2_dr=8

  # RX2 frequency
  #
  # When set to -1, the default RX2 frequency will be used.
```

```
#
# Please consult the LoRaWAN Regional Parameters specification for valid
# options of the configured network_server.band.name.
rx2_frequency=923300000

# Enable only a given sub-set of channels
#
# Use this when ony a sub-set of the by default enabled channels are being
# used. For example when only using the first 8 channels of the US band.
#
# Example:
# enabled_uplink_channels=[0, 1, 2, 3, 4, 5, 6, 7]
enabled_uplink_channels=[0,1,2,3,4,5,6,7]


# Extra channel configuration.
#
# Use this for LoRaWAN regions where it is possible to extend the by default
# available channels with additional channels (e.g. the EU band).
# The first 5 channels will be configured as part of the OTAA join-response
# (using the CFList field).
# The other channels (or channel / data-rate changes) will be (re)configured
# using the NewChannelReq mac-command.
#
# Example:
# [[network_server.network_settings.extra_channels]]
# frequency=867100000
# min_dr=0
# max_dr=5

# [[network_server.network_settings.extra_channels]]
# frequency=867300000
# min_dr=0
# max_dr=5

# [[network_server.network_settings.extra_channels]]
# frequency=867500000
# min_dr=0
# max_dr=5

# [[network_server.network_settings.extra_channels]]
# frequency=867700000
# min_dr=0
# max_dr=5

# [[network_server.network_settings.extra_channels]]
# frequency=867900000
# min_dr=0
# max_dr=5
# Class B settings
[network_server.network_settings.class_b]
# Ping-slot data-rate.
ping_slot_dr=0

# Ping-slot frequency (Hz)
#
# Set this to 0 to use the default frequency plan for the configured region
# (which could be frequency hopping).
ping_slot_frequency=0


# Network-server API
#
# This is the network-server API that is used by LoRa App Server or other
# custom components interacting with LoRa Server.
[network_server.api]
# ip:port to bind the api server
bind="0.0.0.0:8000"

# ca certificate used by the api server (optional)
ca_cert=""

# tls certificate used by the api server (optional)
tls_cert=""

# tls key used by the api server (optional)
tls_key=""

# Gateway API
#
# This API is used by the LoRa Channel Manager component to fetch
# channel configuration.
[network_server.gateway.api]
# ip:port to bind the api server
bind="0.0.0.0:8002"

# CA certificate used by the api server (optional)
ca_cert=""

# tls certificate used by the api server (optional)
tls_cert=""

# tls key used by the api server (optional)
tls_key=""

# JWT secret used by the gateway api server for gateway authentication / authorization
jwt_secret="verysecret"
```

```
# Gateway statistics settings.
[network_server.gateway.stats]
# Create non-existing gateways on receiving of stats
#
# When set to true, LoRa Server will create the gateway when it receives
# statistics for a gateway that does not yet exist.
create_gateway_on_stats=true

# Aggregation timezone
#
# This timezone is used for correctly aggregating the statistics (for example
# 'Europe/Amsterdam').
# To get the list of supported timezones by your PostgreSQL database,
# execute the following SQL query:
#   select * from pg_timezone_names;
# When left blank, the default timezone of your database will be used.
timezone=""

# Aggregation intervals to use for aggregating the gateway stats
#
# Valid options: second, minute, hour, day, week, month, quarter, year.
# When left empty, no statistics will be stored in the database.
# Note, LoRa App Server expects at least "minute", "day", "hour"!
aggregation_intervals=["minute", "hour", "day"]


# MQTT gateway backend settings.
#
# This is the backend communicating with the LoRa gateways over a MQTT broker.
[network_server.gateway.backend.mqtt]
# MQTT topic templates for the different MQTT topics.
#
# The meaning of these topics are documented at:
# https://docs.loraserver.io/lora-gateway-bridge/use/data/
#
# The default values match the default expected configuration of the
# LoRa Gateway Bridge MQTT backend. Therefore only change these values when
# absolutely needed.
# Use "{{ .MAC }}" as an substitution for the LoRa gateway MAC.
uplink_topic_template="gateway/+/rx"
downlink_topic_template="gateway/{{ .MAC }}/tx"
stats_topic_template="gateway/+/stats"
ack_topic_template="gateway/+/ack"

# MQTT server (e.g. scheme://host:port where scheme is tcp, ssl or ws)
server="tcp://mosquitto:1883"

# Connect with the given username (optional)
username=""
```

```
  # Connect with the given password (optional)
  password=""

  # Quality of service level
  #
  # 0: at most once
  # 1: at least once
  # 2: exactly once
  #
  # Note: an increase of this value will decrease the performance.
  # For more information:
https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels
  qos=0

  # Clean session
  #
  # Set the "clean session" flag in the connect message when this client
  # connects to an MQTT broker. By setting this flag you are indicating
  # that no messages saved by the broker for this client should be delivered.
  clean_session=true

  # Client ID
  #
  # Set the client id to be used by this client when connecting to the MQTT
  # broker. A client id must be no longer than 23 characters. When left blank,
  # a random id will be generated. This requires clean_session=true.
  client_id=""

  # CA certificate file (optional)
  #
  # Use this when setting up a secure connection (when server uses ssl://...)
  # but the certificate used by the server is not trusted by any CA certificate
  # on the server (e.g. when self generated).
  ca_cert=""

  # TLS certificate file (optional)
  tls_cert=""

  # TLS key file (optional)
  tls_key=""


# Default join-server settings.
[join_server.default]
# hostname:port of the default join-server
#
# This API is provided by LoRa App Server.
server="http://appserver:8003"

# ca certificate used by the default join-server client (optional)
```

```
ca_cert=""

# tls certificate used by the default join-server client (optional)
tls_cert=""

# tls key used by the default join-server client (optional)
tls_key=""


# Network-controller configuration.
[network_controller]
# hostname:port of the network-controller api server (optional)
server=""

# ca certificate used by the network-controller client (optional)
ca_cert=""

# tls certificate used by the network-controller client (optional)
tls_cert=""

# tls key used by the network-controller client (optional)
tls_key=""
```

Configurações no arquivo docker-compose.yml

```
version: "2"

services:
  loraserver:
    image: loraserver/loraserver:1
    volumes:
      - ./configuration/loraserver:/etc/loraserver

  appserver:
    image: loraserver/lora-app-server:1
    ports:
      - 8080:8080
    volumes:
      - ./configuration/lora-app-server:/etc/lora-app-server

  gatewaybridge:
    image: loraserver/lora-gateway-bridge:2
    ports:
      - 1700:1700/udp
    volumes:
      - ./configuration/lora-gateway-bridge:/etc/lora-gateway-bridge
```

```
redis:
  ports:
    - 6379
  image: redis:4-alpine
  volumes:
    - ./data/redis:/data

mosquitto:
  image: eclipse-mosquitto
  ports:
    - 1883:1883
```

Se houver uma pasta postgres dentro da pasta configurations, pode apagar sem problemas pois a conexão com o banco é remota e não levantada através da docker.